



SYSTEM TABLES

AGPS is used to issue requisitions, solicitations, purchase orders and State Purchasing Contracts and to provide a method to encumber funds and process payments. CFMS is used as a management tool to track professional, personal, consulting and social services contracts and provides a method to encumber funds and process payments. Although there is a functional distinction between AGPS and CFMS and we generally think of them as two separate systems, they are actually one and the same. A good example of this is the KONT table. The KONT screen is the header screen for AGPS contracts issued by State Purchasing. The KENT screen is the header screen for CFMS contracts issued by the agencies. Information displayed on both the KONT and KENT screens is stored in the KONT table.

AGPS KONT SCREEN

ENTER FUNCTION:		TRANS: KONT		DATE:
CONTRACT HEADER TABLE #1				TIME:
KEY IS CONTRACT NUMBER				TERM:
CONTRACT NUMBER...	:			
STATUS CODE.....	:			
STATUS CHANGE DATE:		AWARD DATE..:		FISCAL YR:
VENDOR NUMBER.....	:			
DOCUMENT TYPE.....	:			
TYPE CONTRACT.....	:			REPROCUREMENT:
PURCHASING AGENCY..:	:			
BUYER-CODE.....	:			REPROC SETUP.:
AGENCY NUMBER.....	:			
T-NUMBER.....	:			WHIM.....:
SOLICITATION CODE..:	:			METH OF OPER.:
AWARD CODE.....	:			ITEM RPT IND.:
ORDERING STATUS....	:			BRAND RPT IND.:
				PRIME VENDOR.:
ORIGINAL:	BEG CON	BEG ORDR	END ORDR	END CON
REVISED.:				NOA ISSUE ACT NOA

KONT TABLE

Selection List:  	
is01pzpd.agpskont	Type
acct_rqd	char(1)
acknowledge_date	date
added_by_ocr	char(1)
agcy_cont_num_kont	char(15)
agcy_date_rcvd	date
agency_number_kont	char(6)
agency_req_nbr_kon	char(10)
alt_del_sched_kont	char(1)
award_date	date
award_vend_nbr_kon	char(11)
base_cont_amt	decimal(13,2)
begin_date	date
bgn_pay_date	date
bond_fund_amt	decimal(13,2)
building_id	char(10)
buyer_code_kont	char(3)
cont_approp	char(1)
cont_rev_number	char(9)
contact_person	char(25)
contract_number_ko	char(10)
coop_procure_code	char(2)
cum_amount_order	decimal(13,2)
cum_complaint	char(6)
cum_nbr_cros	char(6)
d_comp_bill_basis	char(1)
date_apvd	date

CFMS KENT SCREEN

ENTER FUNCTION:		TRANS: KENT		DATE:
CONTRACT ENTRY SCREEN				TIME:
KEY IS CONTRACT NUMBER				TERM:
CONTRACT NUMBER:	CONT REV #:	AGCY CONT #:		
CONT		BASE CONT AMT:		
TITL/		REMAINING-FY.:		
DESC		ORIG CONT AMT:		
		# AMEND: CLASS/SUB:		
		NET AMD AMT:		
STATUS CODE..:	:			
DOCUMENT TYPE:	:			
VENDOR/CONTRACTOR #:	:			ORG TYPE:
CONTRACTING AGENCY.:	:			
CONTRACT USER.....	:			
CONT USER SUB-AGCY.:	:			
CONTRACT OFFICER....	:			
T-NUMBER...:	:	LAST BATCH #:		
		BUILDING ID.:		
FY: HOLD PYMT:	ROLL:	ACCT RQD:	CONT APPROP:	SC CHG:
DA: BEG CONT END CONT ENTRD:		CONT LINES:	PRIOR #:	
ORIG:	AGCY RCVD:	OCR RCVD:	APVD:	
REV.:	BUDGET...:	LEGAL...:	RFP.:	
PERF CODE:	RCVD PERF RPT:	NTC ISSUED:	PRNTD PERF:	

CLOSE ENCOUNTERS OF THE THIRD KIND (InfoMaker for AGPS/CFMS)

There are quite a few instances where the screen name and the table name are different in AGPS and CFMS, as with the KENT screen and the KONT table. However, in some cases the screen name will be the same as the table name, such as data entered on the OREC screen in AGPS is stored in the OREC table.

AGPS OREC SCREEN

ENTER FUNCTION:		TRANS: OREC		DATE:	
ORDER RECEIVER HEADER TABLE				TIME:	
KEY IS ORDER NUMBER AND RECEIVER NUMBER				TERM:	
ORDER NO.....:		:			
RECEIVER NO.....:		:			
REQ-AGENCY.....:		:			
DOC TYPE.....:		:	AGCY RECEIVER #:		
SHIP-TO AGENCY.:		-	CONTRACT.:		
STATUS CODE.....:		:	CHG DATE:		
VENDOR INV NO.:		:			
VENDOR NO.....:		:			
ADDRESS 1.....:					
ADDRESS 2.....:					
CITY.....:		STATE:		ZIP:	
RECEIPT DATE.:		PARTIAL/FINAL.:		QTY VARIANCE.:	
CARRIER.....:		WAREHOUSE CODE.:			
COMMENTS.....:					
USERID LAST CHG:		DATE LAST CHANGED:			

OREC TABLE

Selection List: <input type="button" value="Left"/> <input type="button" value="Right"/>	
is01pzpd.agpsorec	Type
agcy_receiver_num	char(25)
carrier_orec	char(20)
comments_orec	char(40)
concatkey	char(13)
contract_no_orec	char(10)
date_last_chg_orec	char(6)
fillerx	char(29)
last_batch_num_ore	decimal(8,0)
order_number_orec	char(10)
order_receiver_num	char(3)
partial_final_ind	char(1)
rec_all_in_one	char(1)
receipt_date_orec	char(6)

The AGPS/CFMS System Tables Quick Reference Card was developed to assist users in selecting the appropriate system tables when creating AGPS/CFMS adhoc reports. A cross reference is given for each of the system screens showing the system table where the data is stored.

FIELDS

IN GENERAL

Identifying which field to use can sometimes be a real challenge. Usually the field name in the table will be similar to the name on the screen. Sometimes the words that make up the field name in the table are in a different order than they appear on the screen, such as ORDER NUMBER on ORD4, the ORDR table field name is NUMBER_ORDR. Also, it may be abbreviated on the screen or in the table, depending on space constraints. There may be some cases where you'll have to give it your best guess.

For example BFY (Budget Fiscal Year) which appears on the ORD4 screen in AGPS. The actual ORDR table field name is ACCTNG-FISCAL-YEAR.

AGPS ORD4 SCREEN

ENTER FUNCTION:		TRANS: ORD4	
ORDER TABLE #4	KEY IS ORDER NUMBER	DT:	TM:
ORDER NUMBER.....	:		TR:
FILE NUMBER.....	:		
STATUS CODE.....	:		ACCT RQD:
DOCUMENT TYPE.....	:		FY ROLL..:
AGCY REQ NUMBER.....	:	BFY:	CONF ORDR:
VENDOR NUMBER.....	:	FY:	PRNT FLAG..:
BILL TO AGY/SUB-AGY..	-	:	
SHIP TO AGY/SUB-AGY..	-	:	
CONTACT PERSON.....			PH:
DATE QUOTE RECVD....		COMMODITY:	CONTRACT.:
DISCOUNT TERMS.....		PCT:	% DAYS..:
DELIVERY TERMS.....		DAYS ARO:	WEEKS ARO:
PURCHASING AGENCY....	:		FAX:
REQUISITION AGENCY...	:		ENC STAT:
BUYER CODE.....	:		WHIM..:
T-NUMBER.....	:		W/H CODE:
RECEIPT DATE.....		PER PAY:	FREQUENCY:
BEGIN AUTH DATE.....		END PAY DATE:	DAY TO PAY:
SCHED PAY DATE 1.....		SCHED PAY DATE 2:	AUTO BATCH PAY:
TOTAL AMOUNT.....		TAX AMOUNT:	LAST PAID:
			NO LINES:

ORDR TABLE

Selection List: <input type="text"/>	
is01pzpd.agpsordr	Type
a_type_counter_ord	decimal(7,0)
acctq_required	char(1)
acctng_fiscal_year	char(2)
address_line1_ordr	char(25)
address_line2_ordr	char(25)
agcy_tax_exempt_nb	char(10)
agency_req_nbr_o	char(10)
alt_del_sched_ordr	char(1)
amt_paid_ordr	decimal(13,2)
auto_batch_ind	char(1)
award_code	char(2)
award_date_ordr	char(6)

Keep an open mind when trying to identify fields. Sometimes narrowing down the search will help. This is one way to do that:

1. Choose a specific record in the system.
2. Enter a WHERE Statement to retrieve only the record chosen (i.e. order #3000004).
3. Select the field(s) to appear on the report.
4. Do not change the field name labels while in Design, this will give you the true name of the field as it appears on in the table window.
5. Run the Report and compare the contents of the field(s) on the report to the record in the system.

LABEL FIELDS

Some tables contain LABEL fields. Label fields are where the name of the field is not hard coded into the program, but can be changed. The value of some label fields can only be changed by a control agency, these are located in the RACG; OACG; OCAC; OFST; OAM2; OPAG; KACG; KCAC and KPAG tables.

For example, the label fields contained on the OACG table identify the different accounting fields and they are FUND, ORG, APROP UNIT, OBJECT, SUB-OBJECT, BS ACCOUNT and REPORT CAT.

OACG SCREEN

ENTER FUNCTION:		TRANS: OACG		DATE:
ORDER ACCOUNTING DISTRIBUTION TABLE				TIME:
KEY IS ORDER NUMBER AND ACCOUNT DIST NUMBER				TERM:
ORDER NUMBER.....:	:			
ACCOUNT DIST NUMBER.:	FY.:			
PAYING AGENCY.....:	:	LA HEALTH CARE TEST REGION		
STATUS CODE.....:	:	PASSED ACCOUNTING PROCESSING		
STATUS CHANGE DATE.:	DT SENT TO ACTG:	AT.:	ACCTS	
ACTION CODE.....:	:	CREATE ACCOUNTING DISTRIBUTION		
AMOUNT.:	FROM LINE:	TO LINE:	GFS BATCH #:	
CHG ENC.....:	NET ENC.....:	AWARDED:		
FUND :	ORG :			
APROP UNIT :	JOB NUMBER :			
OBJECT :	REASON CODES			
SUB-OBJECT :	1 :	:		
BS ACCOUNT :	2 :	:		
REPORT CAT :	3 :	:		

There is a list of the current values for these label fields on the quick reference card. But if the quick reference card is not handy, you can inquire the BLBL screen and it will display the current value.

BLBL SCREEN

ENTER FUNCTION:		TRANS: BLBL		DATE:
ACCOUNTING DISTRIBUTION LABEL TABLE				TIME:
KEY IS DEPT FINANCIAL				TERM:
DEPT FINANCIAL.....:	CNTRL :	GFS		
LABEL 1.....:	FUND			
LABEL 2.....:	ORG			
LABEL 3.....:				
LABEL 4.....:				
LABEL 5.....:	APROP UNIT			
LABEL 6.....:	JOB NUMBER			
OBJECT CODE.....:	OBJECT			
SUB-OBJECT CODE....:	SUB-OBJECT			
LABEL 9.....:	BS ACCOUNT			
LABEL 10.....:	REPORT CAT			

When AFS is implemented in April of 1998, there will be a new label. The value of Label Field 4 will be ACTIVITY.

When looking at the tables in Infomaker, the label fields are easy to spot. They are AGCY_FLD_1 through AGCY_FLD_10 and CNTRL_FLD_1 through CNTRL_FLD_10.

There are some tables that contain label fields where the agencies can change the value of the label. These tables are KONT and KLDM. The label fields on these tables are specific to an AGENCY, FISCAL YEAR and DOCUMENT TYPE. To find the value of the label fields for your agency (if they are used) for KONT inquire KLBL, and for KLDM inquire KLDL.

DATE FIELDS

OSIS is currently preparing AGPS/CFMS for year 2000 compliance, which is separate from the GFS conversion . This project is being implemented in 4 phases. Phase I consisted of conversion of Contract Tables, Phase II Requisition and Solicitation Tables. Both Phase I and II have already been moved into production. Phase III will consist of Order Tables and is scheduled to be moved to production later this year. In Phase I, II and III not all tables in the 'category' were included. Tables where the Fiscal Year is KEY, were not converted. These will be included in Phase IV along with Agency, System, History, Approval, Commodity, Text and Vendor Tables.

In this conversion the date fields are being changed from a ***Character*** data type to a ***Date*** data type. To the user, this change is transparent and has no affect on the way dates are entered into the system, it simply changes the way the dates are stored in the database. However, it does have an impact when using Infomaker to report on date fields. It changes the way a WHERE Statement or FILTER is written and changes how dates appear on reports.

Typically a date stored as a *character* data type will appear on a report exactly as it's stored in the database. For instance February 26, 1998 will appear on reports as 022698. After the conversion to *date* data type the date February 26, 1998 will appear on reports as 02/26/98. The date data type will allow the user while in DESIGN to change the display format to any valid date format i.e., YYYY/MM/DD; DD/MM/YYYY etc.

Additionally, because DB2 requires that all date fields be populated, when the conversion takes place any BLANK date fields in the table will be populated with 00010101 (January 1, 0001) and will appear on reports as 1/1/01.

VERIFY THE DATA TYPE of a date field before using in a WHERE Statement or a FILTER. Data types can be displayed in SQL Select to 'turn on' this feature, while in SQL Select click Design, highlight Show, click Data types. The data type for all fields will be displayed in the table window along with the number of characters contained in the field. A character data type will display CHAR; a date data type will display DATE.

Character Data Type

Dates that are stored as a character data type are cumbersome to use because they are stored as a 'whole' number. They are identified as CHAR(6) in the table window. CHAR represents the data type and (6) indicates the number of characters contained in the field. It is important to identify the order in which the date is stored MMDDYY or YYMMDD because this will determine how the date is used in a WHERE Statement or FILTER. Refer to the AGPS/CFMS System Tablay to identify the correct order.

CLOSE ENCOUNTERS OF THE THIRD KIND (InfoMaker for AGPS/CFMS)

Each character has a position within the field, for example, July 1, 1997:

Position within the field:	<u>1st</u>	<u>2nd</u>	<u>3rd</u>	<u>4th</u>	<u>5th</u>	<u>6th</u>
If stored YYMMDD:	9	7	0	7	0	1
If stored MMDDYY:	0	7	0	1	9	7

* YYMMDD: Can be used in a WHERE Statement or FILTER to retrieve a specific date or a range of dates without using the substring function.

* MMDDYY: Can be used in a WHERE Statement or FILTER to retrieve a specific date, however if a range of dates is needed this would not work. When the date is stored as a whole number if the WHERE Statement is written as follows all records with a date greater than or equal to 070197 would be retrieved. This would include records (if they exist) with dates ranging from 070295-123195; 070296-123196 and 070197-123197. Unless a record existed greater than or equal to 070198, no records for 1998 would be retrieved because as whole numbers 010198 through 063098 are less than 070197:

Column	Oper	Value	Logical
is01pzpd.agpsopay.due_date_opay	>=	'070197'	

To retrieve a range of dates the substring function would be needed. Using the substring identifies SPECIFIC data elements (or characters) within a field. How a substring will help, it will allow the year to be "removed" from the number and the month and day to be used separately in the WHERE Statement.

This will retrieve a range of dates within the SAME calendar year July 1, 1997 through December 31, 1997:

Column	Oper	Value	Logical
(substr(is01pzpd.agpsopay.due_date_opay,5,2)	=	'97'	And
substr(is01pzpd.agpsopay.due_date_opay,1,4)	>=	'0701'	

This will retrieve a range of dates with different calendar years July 1, 1997 through February 26, 1998:

Column	Oper	Value	Logical
(substr(is01pzpd.agpsopay.due_date_opay,5,2)	=	'97'	And
substr(is01pzpd.agpsopay.due_date_opay,1,4)	>=	'0701'	Or
(substr(is01pzpd.agpsopay.due_date_opay,5,2)	=	'98'	And
substr(is01pzpd.agpsopay.due_date_opay,1,4)	>=	'0101'	

Date Data Type

Dates that are stored as a Date Data Type are somewhat easier to use. They are identified as Date in the table window. With the Date data type, the substring function is not necessary. Keep in mind that any BLANK dates are stored as 01/01/0001. There are several formats to choose from when using the date() function in a WHERE Statement or FILTER, they are:

```
date(YMMDD)
date(YYYYMMDD)
date('YYYY/MM/DD')
date('YYYY-MM-DD')
date('YY-MM-DD')
date('YY/MM/DD')
date('YMMDD')
```

Using equal to (=); equal to or greater than (>=); or greater than (>) operators requires only a single line WHERE Statement. For instance, to retrieve a range of dates from July 1, 1997 through February 26, 1998:

Column	Oper	Value	Logical
is01pzpd.agpskont.end_date	>=	date('1997-07-01')	

Using less than (<); equal to or less than (<=); or not equal (<>) requires only a single line WHERE Statement, provided the date field has no BLANKS. However, if records exist where the date is BLANK and are not wanted on the report, another line will need to be written to eliminate them. For instance to retrieve records prior to February 26, 1998, eliminating records with BLANK dates:

Column	Oper	Value	Logical
(is01pzpd.agpskont.end_date	<=	date('1997-02-26')	And
is01pzpd.agpskont.end_date	<>	date('0001-01-01')	

There may be some instances where a range of dates is needed for more than one field. For instance to do a report that will provide a list of all CFMS contracts that are currently active on a specific date. The fields that would be used are Begin Date; Revised Begin Date; End Date and Revised End Date. A comparison of the dates would have to be made for every possible combination. Following is an example of the WHERE Statement that would be written. This statement was written with a retrieval argument (:CURRENT_DATE), so when the report is run the user is prompted to provide the date desired, instead of having to re-write the WHERE Statement in order to change the date:

CLOSE ENCOUNTERS OF THE THIRD KIND (InfoMaker for AGPS/CFMS)

```
WHERE (((IS01PZPD.AGPSKONT.BEGIN_DATE <= :CURRENT_DATE AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE = date('0001-01-01') AND
IS01PZPD.AGPSKONT.END_DATE >= :CURRENT_DATE AND
IS01PZPD.AGPSKONT.REVISED_END_DATE = date('0001-01-01')) OR
((IS01PZPD.AGPSKONT.BEGIN_DATE <> IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE) AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE <> date('0001-01-01') AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE <= :CURRENT_DATE AND
IS01PZPD.AGPSKONT.END_DATE >= :CURRENT_DATE AND
IS01PZPD.AGPSKONT.REVISED_END_DATE = date('0001-01-01')) OR
(IS01PZPD.AGPSKONT.BEGIN_DATE <= :CURRENT_DATE AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE = date('0001-01-01') AND
(IS01PZPD.AGPSKONT.END_DATE <> IS01PZPD.AGPSKONT.REVISED_END_DATE) AND
IS01PZPD.AGPSKONT.REVISED_END_DATE <> date('0001-01-01') AND
IS01PZPD.AGPSKONT.REVISED_END_DATE >= :CURRENT_DATE) OR
((IS01PZPD.AGPSKONT.BEGIN_DATE <> IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE) AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE <> date('0001-01-01') AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE <= :CURRENT_DATE AND
(IS01PZPD.AGPSKONT.END_DATE <> IS01PZPD.AGPSKONT.REVISED_END_DATE) AND
IS01PZPD.AGPSKONT.REVISED_END_DATE <> date('0001-01-01') AND
IS01PZPD.AGPSKONT.REVISED_END_DATE >= :CURRENT_DATE) OR
(IS01PZPD.AGPSKONT.BEGIN_DATE <= :CURRENT_DATE AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE = date('0001-01-01') AND
(IS01PZPD.AGPSKONT.END_DATE = IS01PZPD.AGPSKONT.REVISED_END_DATE) AND
IS01PZPD.AGPSKONT.REVISED_END_DATE <> date('0001-01-01') AND
IS01PZPD.AGPSKONT.REVISED_END_DATE >= :CURRENT_DATE) OR
((IS01PZPD.AGPSKONT.BEGIN_DATE = IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE) AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE <> date('0001-01-01') AND
IS01PZPD.AGPSKONT.REVISED_BEGIN_DATE <= :CURRENT_DATE AND
IS01PZPD.AGPSKONT.END_DATE = IS01PZPD.AGPSKONT.REVISED_END_DATE) AND
IS01PZPD.AGPSKONT.REVISED_END_DATE <> date('0001-01-01') AND
IS01PZPD.AGPSKONT.REVISED_END_DATE >= :CURRENT_DATE))))
```

Don't Panic. This particular report is available on the Internet at the OSIS/AdHoc Reporting Home Page., the name is CFMS1M.PBL and CFMS1CI.PBL. The report can be run by downloading the pbl file from the Internet, logging in to Infomaker, signing on to production and previewing the report.

***** VERY IMPORTANT *****

BEFORE DOWNLOADING PBL FILES READ THE INSTRUCTIONS ON THE ADHOC REPORTING HOME PAGE

AND

READ THE REPORT DESCRIPTIONS, THEY WILL TELL YOU IF CHANGES CAN BE MADE TO THE REPORT

ARCHIVE TABLES

Archive data tables are used exactly as production data tables. They contain the same fields and field formats as their production counterparts. The archive tables are identified by IS05PZHD.ARCH____(in the old Eda Server PRODPRVT) and EDADBA.ARCH____(in the new Eda Server EDASQLP). Archive tables can be joined with other Archive tables or Production tables.

When records are archived only the information stored on the table is archived. To further explain, when a purchase order or contract is entered into the system the user enters the vendor number. On some screens when the order/contract is viewed on-line the vendor name is displayed. The vendor name is not stored in the ORDR/KONT table, it is 'joined' to the VEND table by vendor number. So, in order to do a report from the archive tables for an order or contract and have the vendor name appear on the report the ORDR/KONT(archive table) would have to be joined with VEND(Production table).

ENTER FUNCTION:		TRANS: ORD4	
ORDER TABLE #4	KEY IS	ORDER NUMBER	DT: TM: TR:
ORDER NUMBER.....:	:	:	:
FILE NUMBER.....:	:	:	:
STATUS CODE.....:	:	:	ACCT RQD:
DOCUMENT TYPE.....:	:	:	FY ROLL.:
AGCY REQ NUMBER.....:	:	BFY: FY: CONF ORDR: PRNT FLAG..:	:
VENDOR NUMBER.....:	160484732 00	:	OCIDENTAL CHEMICAL CORP.
BILL TO AGY/SUB-AGY.:	-	:	:
SHIP TO AGY/SUB-AGY.:	-	:	:
CONTACT PERSON.....:	:	:	PH:
DATE QUOTE RECVD.....:	:	COMMODITY:	CONTRACT.:
DISCOUNT TERMS.....:	:	PCT: % DAYS.:	NET:
DELIVERY TERMS.....:	:	DAYS ARO:	WEEKS ARO:
PURCHASING AGENCY...:	:	:	FAX:
REQUISITION AGENCY...:	:	:	ENC STAT:
BUYER CODE.....:	:	:	WHIM.:
T-NUMBER.....:	:	:	W/H CODE:
RECEIPT DATE.....:	:	PER PAY: FREQUENCY: DAY TO PAY:	:
BEGIN AUTH DATE.....:	:	END PAY DATE:	AUTO BATCH PAY:
SCHED PAY DATE 1....:	:	SCHED PAY DATE 2:	LAST PAID:
TOTAL AMOUNT.....:	:	TAX AMOUNT:	NO LINES:

Selection List: <input type="button" value="Left"/> <input type="button" value="Right"/>	
IS05PZHD.ARCHORDR	Type
TYPE_WAIVER_ORDR	char(1)
USERID_LAST_CHG_OR	char(8)
VEND_AAS_STATUS	char(1)
VEND_BID_REF_NUMBE	char(14)
VENDOR_NUMBER_ORDR	char(11)
WAIVER_NUMBER	char(10)
WAREHOUSE_CODE_ORD	char(4)
WEEKS_ARO_ORDR	char(3)
WHIM_ORDR	char(1)

Selection List: <input type="button" value="Left"/> <input type="button" value="Right"/>	
IS01PZPD.AGPSVEND	Type
MTD_AMT_CHANGED	decimal(13,2)
MTD_AWARD_COUNT	decimal(7,0)
MTD_CHANGE_COUNT	decimal(7,0)
NAME1	char(30)
NAME2	char(30)
NBR_ACT_COMP_CODE1	decimal(6,0)
NBR_ACT_COMP_CODE2	decimal(6,0)
NBR_ACT_COMP_CODE3	decimal(6,0)
NBR_ACT_COMP_CODE4	decimal(6,0)

Archive / Production Text Tables

As discussed, archive tables work exactly like production tables. Which is also true for text tables. In production everything can be viewed on-line, but that's not the case with archived records.

How the text tables in AGPS/CFMS work: Text tables in AGPS/CFMS can become EXTREMELY large, because each line of text is stored as a separate record. For example, an OMOD description containing 40 lines of text would be stored as 40 records. How the system minimizes the unnecessary use of space is that each commodity number entered into AGPS/CFMS is given a commodity description which is found on the CSPC table. When a buyer enters a purchase order in AGPS, but does not enter a description, the commodity description from the CSPC table is 'joined' to the order line by the commodity number. Although this description appears on the OFST/OLI4 screens, unless a change is made by the buyer it is never written to the OMOD table. The same is true for contract line descriptions, entered by State Purchasing on the KMOD table. If the buyer adds the line to an order, but does not make changes, the contract line description from the KMOD table is 'joined' to the order line by contract and contract line number. By making these 'joins' instead of writing the description to OMOD saves tremendous space.

How this affects reporting in Infomaker: If a report is wanted showing the order line description, and the OMOD was never written, no description would be retrieved. Not being sure of where the description is stored, it's best to retrieve all possibilities. For Non-Contract orders, you would retrieve CSPC and OMOD. If CSPC and OMOD both appear changes were made and an OMOD was written. However, if OMOD does not appear no changes were made to the CSPC description. By retrieving both you are assured an item description. Contract Release Orders are a little more involved, because some of the State Purchasing contracts have been archived. For Contract Release Orders, you would retrieve OMOD, KMOD-Production and KMOD-Archive. Both KMOD-Production and KMOD-Archive are needed unless you're sure of the location of the contract. If OMOD and KMOD appear, changes were made and an OMOD was written. However, if OMOD does not appear, no changes were made to the contract description from KMOD-Production or KMOD-Archive.

How to work around this: NESTED REPORTS. A Nested Report is a report that is 'retrieved' into a Base Report. The Nested Report is written with a retrieval argument, which is supplied by the Base Report. One reason nested reports work better than trying to use joins to the CSPC/OMOD/KMOD-Production/KMOD-Archive tables is that joins to text tables will sometimes result in a -905 error because the size of the text tables are so large. Another reason they work better is the concatkey field of the tables can be used for retrieval arguments. The concatkey field indexes the table. This speeds up the retrieval time of the report considerably. Below is a list of these text tables along with the fields that are contained in the concatkey and the length of the concatkey.

TABLE
CSPC

CONCATKEY FIELDS
Class, Sub Class, Item, Text Line # (15)

OMOD	Order #, Order Line #, Text Line # (19)
KMOD	Contract #, Contract Line #, Text Line #(19)

How to do a NESTED REPORT to retrieve an order line description: If it's a Non-Contract order, only CSPC and OMOD (#1 & #2) are needed. If its a Contract order OMOD, KMOD-Production and KMOD-Archive (#2, #3 & #4) are needed. You can create a report and include CSPC, OMOD, KMOD-Production and KMOD-Archive which can be run regardless if it is a Non-Contract or Contract Order.

1. Create a nested report for CSPC:

Using the CSPC table select the fields you want to display on the report. For CSPC choose TEXT_CSPC. Click on DESIGN, highlight and click RETRIEVAL ARGUMENTS. The Specify Retrieval Arguments Window appears. Name the argument (because this is a text table, the concatkey field should be used as the column, the concatkey for CSPC consists of commodity number and text line number) so you could name the argument commodity_number and select a TYPE (this is referring to the Data Type, which was covered earlier. Char Data Type = String) then click OK. Write a WHERE Statement (the base report will supply this retrieval argument once the nested report is placed in the base report):

Column	Operator	Value
is05pzpd.agpscspc.concatkey	like	:commodity_number (when entering VALUE for a retrieval argument, right click and a pop-up appears, highlight and click ARGUMENTS and the retrieval argument that was entered in the Specify Retrieval Arguments Window will be displayed, highlight it and click on PASTE.)

Go to DESIGN then PREVIEW the report. You will be prompted to supply a retrieval argument. Because the concatkey field was used, you will have to enter a CLASS, SUBCLASS and ITEM Number along with a % sign, i.e. 61518000000%. The % sign is a wildcard, by entering the commodity number and % this will retrieve all of the text lines for the commodity, this is needed because text lines are included in the CONCATKEY field of CSPC. Save the Report.

2. Create a nested report for OMOD:

Using the OMOD table select the fields you want to display on the report. For OMOD choose TEXT_OMOD. Click on DESIGN, highlight and click RETRIEVAL ARGUMENTS. The

Specify Retrieval Arguments Window appears. Name the argument (because this is a text table, again the concatkey field should be used as the column, the concatkey for OMOD consists of the order number, order line number and text line number) so you could name the argument line_number and select a TYPE then click OK. Write a WHERE Statement (the base report will supply this retrieval argument once the nested report is placed in the base report):

Column	Operator	Value
is05pzhd.agpsomod.concatkey	like	:line_number

Go to DESIGN then PREVIEW the report. You will be prompted to supply a retrieval argument. Because the concatkey field was used, you will have to enter ORDER Number and the % sign i.e. 3000002%, this is needed because the order, order line and text line numbers are included in the CONCATKEY field of OMOD. Save the Report.

3. Create a nested report for KMOD-Production:

Using the KMOD table from Production select the fields you want to display on the report. For KMOD choose TEXT_KMOD. Click on DESIGN, highlight and click RETRIEVAL ARGUMENTS. The Specify Retrieval Arguments Window appears. Name the argument (because this is a text table, the concatkey field should be used in the retrieval argument as the column, but to use the KMOD concatkey field you need to do a substring to identify only the contract number in the concatkey for KMOD) so you could name the argument contract_number. Another RETRIEVAL ARGUMENT is needed for the contract line number, While the Specify Retrieval Arguments window is open, click ADD, name the new argument, this can be named contract_line and the TYPE should be NUMBER. Also, both arguments will have to be identified separately in the WHERE Statement (the base report will supply both retrieval arguments once the nested report is placed in the base report).

Column	Operator	Value
substr(is01pzpd.agpskmod.concatkey,1,10)	like	:contract_number
is01pzpd.agpskmod.contract_line	=	:contract_line

Go to DESIGN then PREVIEW the report. You will be prompted to supply a retrieval argument, because the concatkey field was used, you will have to enter a contract number and the % sign, i.e. 4000002% and a contract line number. Save the Report.

4. Create a nested report for KMOD-Archive:

Using the KMOD table from Archives complete all steps outlined in #3.

***BOTH KMOD-PRODUCTION AND KMOD-ARCHIVE ARE NEEDED, BECAUSE IF**

THE CONTRACT WAS ARCHIVED, NO DESCRIPTION WOULD BE RETRIEVED IF ONLY THE KMOD-PRODUCTION IS USED.

5. Create a base report:

Using the OLIN table select the fields you want to display on the report, be sure to include contract line number (this is used as a retrieval argument for the nested KMOD report). You can identify the specific order number you want to retrieve in the WHERE statement of the base report or write a retrieval argument so Infomaker will prompt you to provide the order number when the report is run. It is recommended that the concatkey field be used to retrieve regardless if written in the WHERE statement or as an argument. To use in a WHERE statement simply use the LIKE operator and using single quotes follow the order number with a % sign, i.e. '3000002%'. To enter a retrieval argument, click on DESIGN, highlight and click RETRIEVAL ARGUMENTS. The Specify Retrieval Arguments Window appears. Enter a name for the retrieval argument and select a TYPE then click OK. Write a WHERE Statement (either identifying the order number or using a retrieval argument, this example is using a retrieval argument):

Column	Operator	Value
is01pzpd.agpsolin.concatkey	Like	:order_number

The Base Report will require several COMPUTE statements, because we're using the concatkey fields in the nested reports. By writing COMPUTE statements we will 'concatenate' the '%' sign to the fields in the Base Report that will supply retrieval arguments to the Nested Reports:

```
CSPC
  is01pzpd.agpsolin.commodity_nbr_olin || '%'
OMOD
  is01pzpd.agpsolin.concatkey || '%'
KMOD
  is01pzpd.agpsolin.contract_nbr_olin || '%'
(This will be used for KMOD-Production and Archive)
```

****Remember the order in which these COMPUTE statements are written. Infomaker assigns a name and the compute statements appear as columns on the report. The names are assigned in the order they are written.**

Go to DESIGN, this is where the nested reports will be added to the base report. Pull down the Detail Band so that there is plenty of room to work with. Click OBJECTS, highlight REPORT and click. Position the cursor where you want the report to appear and click. The Report Window appears,

highlight the report you want and click OK. Right click on the report *4.0 users highlight and click Retrieval Arguments *5.0 users highlight and click on Properties, then click on Arguments. The Arguments window will identify the Retrieval Argument from the Nested Report. Click on the pull-down arrow (these are all of the columns selected in the base report along with the COMPUTE statements that were written), highlight the appropriate computed field for the nested report (remember when specifying retrieval arguments for KMOD to also include the contract line number). REPEAT FOR EACH NESTED REPORT. PREVIEW the report, enter the order number followed by the % sign. Save the Report.

AVAILABLE ON THE WEB

***** VERY IMPORTANT *****

BEFORE DOWNLOADING PBL FILES READ THE INSTRUCTIONS ON THE ADHOC REPORTING HOME PAGE

AND

READ THE REPORT DESCRIPTIONS, THEY WILL TELL YOU IF CHANGES CAN BE MADE TO THE REPORT

AGPS1CL.PBL Archived Purchase Orders (r_archived_purchase_order)

Created in Infomaker 4.0 - Eda Ident Case = Case Ignored. Additional joins can be made, but do not delete existing WHERE Statements or Computed Fields. Provides the user with a copy of an archived purchase order, including header information, line information along with commodity description, order vendor text and accounting information. Report Displays: Order #, Title, FY, Vendor #, Vendor Name, Doc Type, Order Status, Req Agency, Pur Agency, Bill To, Ship To, Days/Weeks ARO, Agency Ref #, Number Items, Per Pay Indicator, Acct Rqd Indicator, Confirmation Flag, Order Amount, Contact, Contact Phone, Comm Line Number, Line Status, Commodity #, Qty, Unit of Measure, Unit Price, Extended Amount, Contract #, Contract Line #, Line Per Pay Amount, CSPEC/OMOD/KMOD description, Account Distribution, From Line, To Line, Pay Agency, Status, Amount, Amount Processed, Fund, Org, Function, Activity, Apropos Unit, Job Number, Object, Sub-Object, BS Account and Reporting Cat.

When previewing user is prompted to enter seven (7) digit Order Number.

AGPS1M.PBL Archived Purchase Orders (r_archived_purchase_order)

Created in Infomaker 4.0 - Eda Ident Case = Mixable.

****Same As AGPS1CL.PBL****

AGPS2CI.PBL Solicitation Vendor Labels (r_solicitation_vendor_labels)

Created in Infomaker 4.0 - Eda Ident Case = Case Ignored. Additional joins can be made, do not delete existing WHERE Statements. Provides the user with Solicitation Vendors in LABEL format. Set 2 across, 7 down 1.33 x 4.25 labels, but can be changed in DESIGN. Labels Display: Vendor Number, Name, Address type, Address, and Solicitation Number.

When previewing user is prompted to enter seven (7) digit Solicitation Number.

AGPS2M.PBL Solicitation Vendor Labels (r_solicitation_vendor_labels)

Created in Infomaker 4.0 - Eda Ident Case = Mixable.

****Same as AGPS2CI.PBL****

CFMS1CI.PBL Current Active Contracts (r_active_contracts)

Created in Infomaker 4.0 - Eda Ident Case = Case Ignored. Do not delete existing WHERE Statement, DO NOT convert to syntax. By Agency provides the user with a list of all active contracts as of a specific date. Report Displays: Contract #, Agency#, Begin Date, Revised Begin Date, End Date and Revised End Date.

When previewing user is prompted to enter six (6) digit Agency Number (use CAPS if letters are used in Agency Number) and desired date in MM-DD-YY format.

CFMS1M.PBL Current Active Contracts (r_active_contracts)

Created in Infomaker 4.0 - Eda Ident Case=Mixable.

****Same as CFMS1CI.PBL****

CFMS2CI.PBL Contracts to be Encumbered for New Year (r_cfms_contracts_composite)

Created in Infomaker 4.0 - Eda Ident Case = Case Ignored. No changes can be made. By Agency provides the user a list of contracts and amendments to be encumbered for the new year, where the KOFY encumbered flag is not equal E. Report includes new and existing contracts that will not expire on date entered. Report Displays: Contract #, Agency #, Kent End Date, Revised End, KOFY FY, KOFY Enc Status.

When previewing user is prompted to enter six(6) digit Agency Number (use CAPS if letters are used in Agency Number) and Fiscal Year End Date in YYYY-MM-DD format.

CFMS2M.PBL Contracts to be Encumbered for New Year(r_cfms_contracts_composite)

Created in Infomaker 4.0 - Eda Ident Case=Mixable.

****Same as CFMS2CI.PBL****